CLUBE

Z80

Janeiro/84

N.º 16

Coordenadores: MARIA IRENE • ALEXANDRE SOUSA • J. MAGALHÃES • AV. BOAVISTA, 832-2.º T. • TELEF. 65127 • 4100 PORTO

NESTE NÚMERO

POR FALTA DE ESPAÇO, ESTE NÚMERO NÃO INCLUI TODAS AS RÚBRICAS USUAIS, NOMEADAMENTE O ESPAÇO SPECTRUM.
SERÃO PUBLICADAS NO PRÓXIMO NÚMERO.

No Interior:

Cupão de Inscrição

Edição: Clube Z80

Fotocomposição: Fotomecânica Mabreu/Porto

Impressão: Gráfica Firmeza/Porto

Tiragem: 500 exemplares, Janeiro 1984

EDITORIAL

Os nossos apelos às vossas sugestões, comentários e ideias começam a surtir efeito.
Mesmo quando há falta de tempo e/ou preguiça para escrever, há outros meios mais simples. *José Manuel Gorda* (Moncorvo) pegou numa cassete e num gravador e falou-nos assim:

«(...) Dou-vos os mais sinceros parabéns pela apresentação, modo gráfico e conteúdo do boletim. Mas parece-me que não tem sido dados os grandes passos para o engrandecimanto do CLUBE Z 80.

1.° Neste mundo de consumo, a publicidade é essencial:

Que tal a inclusão de 1 pequeno anúncio nos suplementos de informática de alguns jornais e revistas (nomeadamente "Selecções de Rádio")?
Porque não contactar a Timex Portuguesa ou a Landry para juntarem ao material que vendem uma referência ao CLUBE?

2.° (. . .) Criar um clube paralelo, dentro do CLUBE Z 80, só a nível de Software (cassetes e livros) em que o associado entraria com uma verba para prémios e despesas e que funcionaria mais ou menos assim: O sócio interessado entregaria uma quantia pré-estabecida (ex.: 1000$00) (¹) e requisitava os programas que quisesse. Quando os tivesse em seu poder, corrê-los-ia na máquina e se lhe interessassem, teria 2 caminhos a seguir:

— ficava c/ a(s) cassete(s) e pagava ou trocava-as por outras (de = valor ou acrescentaria a diferença) e pagaria uma taxa de troca ou utilização de nova cassete (ex.: 100$00), acrescentando-se os portes de correio.
O mesmo se poderia aplicar aos livros mas com uma taxa de utilização mais baixa (ex.: 50$00).

(¹) Seriam uma "conta-corrente" para registos de correio, taxas de utilização . . . .

No entanto, ficaria estabelecido por escrito que o sócio poderia recorrer a estes serviços se devolvesse o material em bom estado e num prazo a estabelecer . . .
Isto poderá ser um pouco complicado, mas acredito na aceitação pela maioria dos sócios. E acredito plenamente nesta ideia. De mim, tem ao dispor um amigo e só tenho pena de não estar mais perto.

3.° Reservar duas páginas só com títulos e preços (catálogo) em todos os boletins, além de apresentarem descritivamente o que vão adquirindo (. . .)».

Também Hugo Assumpção/Lisboa partilha da 1.ª ideia que registámos anteriormente: "A maioria dos possuidores de micros não tem conhecimento da existência do clube. Há que fazer campanha: Por exemplo, distribuir aos sócios um cartaz que colocariam em escolas, clubes, locais de reunião, etc., de modo a que as pessoas se apercebessem da s/ existência e se inscrevessem".
Correspondendo ainda às n/ solicitações, este sócio enviou-nos alguns programas (que brevemente, serão publicados) e as instruções completas do jogo "MANIC MINER" 48 K (à disposição de quem estiver interessado.

O CLUBE Z 80 agradece e vai tentar pôr em prática estas sugestões. Contudo, deixámo-las também à consideração de todos os sócios, nomeadamente a segunda ideia de *José Gorda,* sobre a qual esperámos os v/ comentários.

Um grande abraço
Alexandre Sousa/J. Magalhães
Maria Irene

O SEU NOME NOS SEUS PROGRAMAS — SPECTRUM

Para impedir o vizinho do lado de escrever o nome onde não deve, coloque o seu nome na linha zero de um programa:

```
1 LET A = PEEK 23637 + 256 * PEEK 23638 : POKE A, 0 : POKE A + 1, 0 : STOP
2 REM (C) JULIA CREST
```

Use a tecla RUN e retire a linha 1. Se não gosta do nome que está na linha 2, substitua-o pelo nome que lhe agrada mais.

INTRODUÇÃO À LINGUAGEM MÁQUINA

ZX81

Autor: FERNANDO PRECES/SACAVÉM

(Continuação)

2.º JOGO — BOLA EM MOVIMENTO

Este exemplo serve para recapitular as etapas necessárias na programação em código máquina e, ao mesmo tempo, para preparar o leitor a encarar com um certo *à-vontade* a construção de rotinas mais extensas do que aquelas que têm sido introduzidas neste texto.

1.ª ETAPA:

Elaboração do programa em BASIC.

```
 20 SLOW
 30 CLS
 40 GO SUB 100
 50 FOR A=1 TO 18
 60 PRINT AT A,0;"▩";AT A,31;"▩"
      (GRAFICO 8)
 70 NEXT A
 80 GO SUB 100
 90 GO TO 140
100 FOR A=0 TO 31
110 PRINT "▩";          (GRAFICO 8)
120 NEXT A
130 RETURN
140 LET ED=1
150 LET CB=1
160 LET X=3
170 LET Y=INT (RND*33+8)
180 PLOT X,Y
190 IF INKEY$="R" THEN RUN
200 IF X=2 OR X=61 THEN LET ED=
-ED
210 IF Y=6 OR Y=41 THEN LET CB=
-CB
220 REM UNPLOT X,Y
230 LET X=X+ED
240 LET Y=Y+CB
250 GO TO 180

RUN...
```

As linhas 20 a 130 dão-nos uma área rectangular para o jogo. O uso de uma linha temporária (135 STOP) pode-nos mostrar somente essa parte do programa.

As linhas 140 a 150, iniciam a direcção da bola. ED é a variável para o movimento horizontal: + 1 para a direita, – 1 para a esquerda. CB é a varável para o movimento vertical: + 1 para cima – 1 para baixo.

As linhas 160 e 170 dão o início dos valores para as variáveis X e Y, usando um comando PLOT. O valor X é pré-fixado e o valor Y é escolhido com RAND (número aleatório).

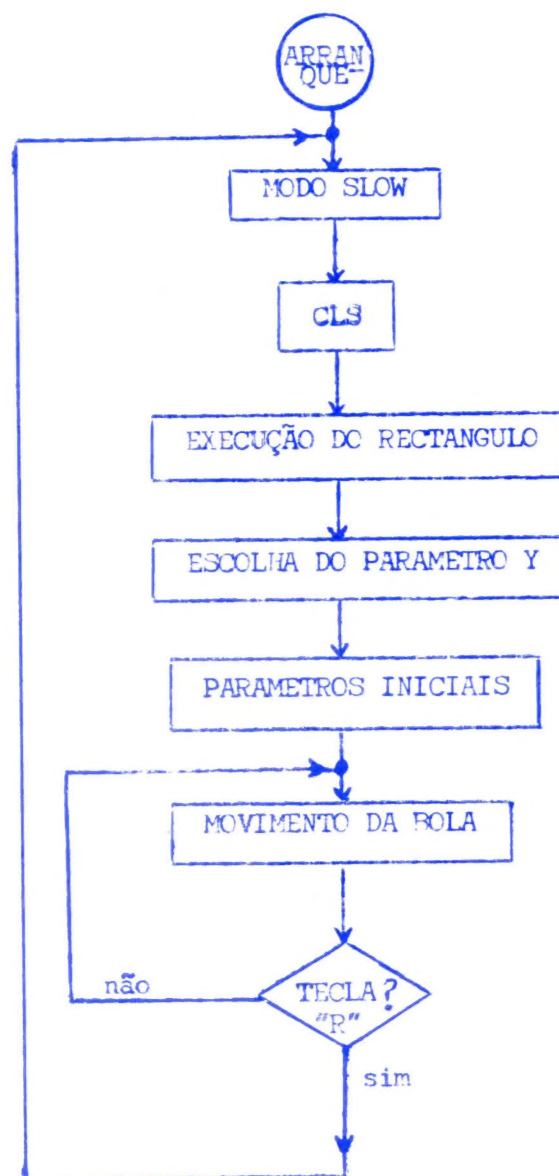
A linha 190 serve para iniciar outro jogo.

As linhas 200 e 210 testam a posição da bola e, se esta toca um dos limites do rectângulo, invertem a sua direcção.

A linha 220 apaga o rasto da bola.

As linhas 230 e 240 dão novos valores para X e Y, no movimento da bola.

2.ª ETAPA:

Diagramas blocos deste programa.

Ao analisarmos o diagrama bloco e o programa em Basic, verifica-se ser possível escrever em código máquina todos os blocos funcionais, com excepção do bloco de escolha do parâmetro Y.

Este utiliza o comando Basic RND que gera um número aleatório em binário de ponto flutuante, que no

ZX81 é guardado em 5 bytes (1 para o expoente e 4 para a matissa), o que torna muito complicada a sua passagem a C.M., e totalmente desaconselhável nesta fase do curso. Partindo desta base, podemos agora estruturar o BASIC necessário ao nosso programa misto, que terá esta configuração:

```
10 REM XXXXXX
15 REM XXXXXX
20 LET K=USR XXXXX
30 POKE XXXXX,INT (RND*33+3)
40 LET K=USR XXXXX
50 RUN
```

Linhas 10 e 15	Programa em código máquina
Linha 20	Executa o modo SLOW, CLS e forma o rectângulo
Linha 30	Escolha do parâmetro Y
Linha 40	Escolha dos parâmetros restantes, movimento da bola e pesquisa da tecla "R".

Quando um programa em Basic está a ser executado, o monitor testa no final de cada linha de instruções se a tecla BREAK é primida e, na afirmativa, interrompe a execução do programa.
Em código máquina a tecla BREAK fica *inativa*, a menos que seja incluída uma dada instrução nessa rotina.
Neste jogo vamos fazer com as que as teclas "R" e "BREAK" sejam testadas. A primeira para recomeçar um jogo e a segunda para interromper o programa. Se não o fizermos, somente paramos a execução do programa, cortando a tensão do ZX.

DIAGRAMA BLOCO PARA A PRIMEIRA ROTINA EM C.M.

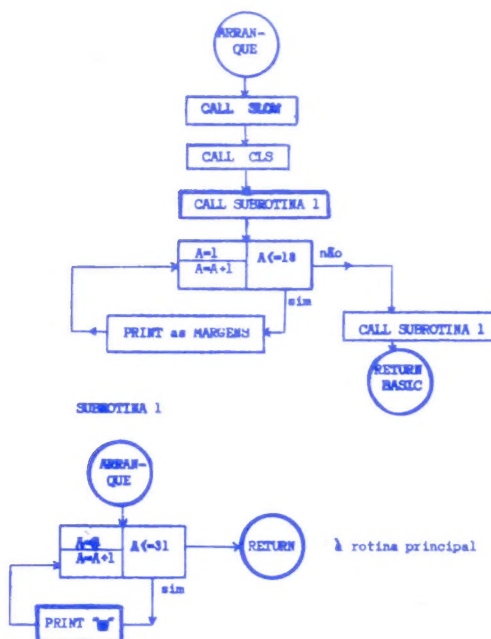
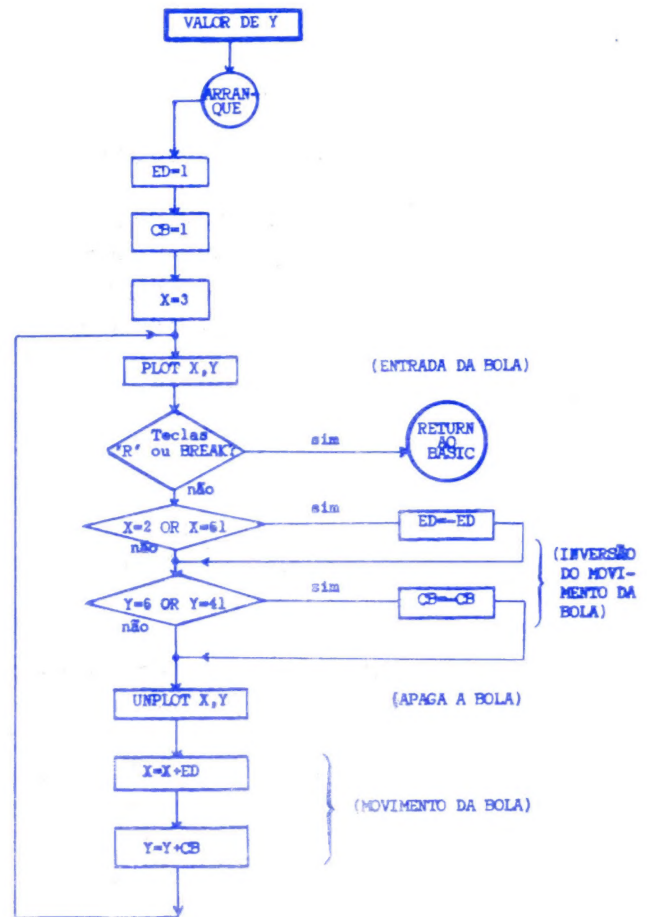



DIAGRAMA BLOCO PARA A SEGUNDA ROTINA EM C.M.

3.ª ETAPA:

Análise dos diagramas blocos e preparação de rotinas em pseudo-Basic, que nos permitam uma aproximação ao "assembler".
Em primeiro lugar vamos elaborar uma subrotina que vá escrever as margens horizontais do rectângulo de jogo.
Precisamos de arranjar uma instrução CICLO que escreva os 32 caracteres (código gráfico 8) *em linha* e a mais indicada será uma instrução DJ NZ. O formato inicial da listagem dessa subrotina em assembler é o seguinte:

ETIQUETA	MNEMÓNICAS	COMENTÁRIOS
	LDA, 8	Caracter a escrever
CONTADOR	LDB, 32	os 32 caracteres
	RST 0016	ROTINA ROM escreva caracter
NEXT	DJNZ	CICLO que lê valor de B, subtrai 1 a B e salta quando B = 0

A rotina principal começa por 2 instruções CALL que provocam o salto tipo GOSUB para 2 rotinas do programa monitor (na ROM).

```
CALL    SLOW
CALL    CLS
```

O bloco para escrever as margens verticais é um pouco mais complicado.

Para uma melhor compreensão deste bloco vamos subdividi-lo para o tornar mais simples. Neste caso, pegando no Basic que executa o dito trabalho, vamos transformá-lo num pseudo Basic e aproximá-lo, de tal forma que ele represente linha a linha o Assembler aceite pelo microprocessador Z 80.

Começemos então:

```
50 FOR A=1 TO 18
60 PRINT AT A,0;"▒";AT A,31;"▒"
70 NEXT A
```

A primeira transformação a efectuar, visto que vamos trabalhar com uma instrução DJ NZ, será a de inverter o ciclo do contador porque o ciclo máquina trabalha com um valor atribuído para zero. O comando PRINT AT terá de ser subdividido em duas partes pois existe uma margem vertical a ser executada na coluna 0, e outra na coluna 31. Assim teremos:

```
50 FOR B=18 TO 1 STEP -1
54 PRINT AT 19-B,0;
60 LET A=8
62 PRINT CHR$ A
66 PRINT AT 19-B,31;
72 LET A=8
74 PRINT CHR$ A
78 NEXT B
```

Note que é muito importante, logo que inicie a transformação do Basic em pseudo Basic, começar por utilizar o *nome* das variáveis identicas às utilizadas nos registos do Z 80.

Este utiliza o registo B como contador e o registo A para a escrita de caracteres.

Como na rotina PRINT AT (já nossa conhecida) o registo B retém o número de linha e o C o número de coluna, temos de incluir também no pseudo Basic uma variável C. Para usar o registo B como variável de linha, estando ele a ser utilizado como contador do ciclo, é necessário salvaguardar o seu conteúdo pelo que vamos utilizar a pilha (STACK), também já nossa conhecida, para armazenar esse valor.

Face a estas novas exigências o pseudo Basic terá de ser transformado para:

50 FOR B = 18 TO 1 STEP − 1
52 LET STACK = B (transfere conteúdo de B temporariamente)
54 LET B = 19 − B (define linha para PRINT AT)
56 LET C = 0 (define coluna para PRINT AT)
58 PRINT AT B,C;
60 LET A = 8 (coloca em A o caracter a ser escrito)

62 PRINT CHR$ A
64 LET B = STACK (traz o valor guardado)
66 LET B = 19 − B (define novo valor da linha para PRINT AT)
68 LET C = 31 (define novo valor da coluna para PRINT AT)

70 PRINT AT B,C;
72 LET A = 8 }
74 PRINT CHR$ A } escrita de caracter gráfico 8
76 LET B = STACK (coloca em B o valor do contador)
78 NEXT B (subtrai 1 a B até B = 0)

Esta última transformação consegue uma aproximação que, à primeira vista, parece ser o retoque final que nos dará o "Assembler".

No entanto vamos observar melhor o que se passa nas linhas 64 e 76.

Na linha 64, porque precisámos do valor do contador em B para efectuarmos a operação da linha 66, fomos buscar ao STACK o valor lá armazenado por acção da linha 52.

Como já sabemos do anterior, logo que se saca um dado do STACK, o seu apontador é incrementado e vai apontar o dado anterior armazenado, e então a linha 76 quando vai buscar o valor do contador ao STACK traz um dado incorrecto. Precisamos pois de remediar a situação, introduzindo na instrução seguinte o mesmo dado de novo no STACK.

Por outro lado não é possível trabalharmos em linguagem máquina com a expressão (LET B = 19 − B) se esta não for subdividida para uma maior aproximação. Teremos de fazer pois um novo arranjo.

PSEUDO BASIC	MNEMÓNICAS CORRESPONDENTES
50 FOR B=18 TO 1 STEP -1	LDB,+N
51 LET STACK=B	PUSH BC
52 LET A=19	LDA,+N
53 LET A=A-B	SUB B
54 LET B=A	LDB,A
56 LET C=0	LDC,+N
58 PRINT AT B,C;	CALL PRINT AT
60 LET A=8	LDA,+N
62 PRINT CHR$ A	RST 16
64 LET B=STACK	POP BC
65 LET STACK=B	PUSH BC
66 LET A=19	LDA,+N
67 LET A=A-B	SUB B
68 LET B=A	LD B,A
69 LET C=31	LD C,+N
70 PRINT AT B,C;	CALL PRINT AT
72 LET A=8	LD A,+N
74 PRINT CHR$ A	RST 16
76 LET B=STACK	POP BC
78 NEXT B	DJNZ (linha 51)

Esta rotina em Pseudo Basic tem agora uma correspondência (um para um) com a rotina a criar em código máquina; porém, como há algumas instruções repetidas que podem ser agrupadas numa sobrotina, podemos melhorar o sistema.

Eis o último arranjo:

```
50 FOR B=18 TO 1 STEP -1          LDB,+N
51 LET STACK=B                    PUSH BC
52 LET C=0                        LDC,+N
53 GO SUB 70                      CALL (addr.)
54 LET B=STACK                    POP BC
55 LET STACK=B                    PUSH BC
56 LET C=31                       LDC,+N
57 GO SUB 70                      CALL (addr.)
58 LET B=STACK                    POP BC
59 NEXT B                         DJNZ (linha 51)
60 RETURN (da rotina)
70 LET A=19                       LDA,+N
71 LET A=A-B                      SUB B
72 LET B=A                        LD B,A
73 PRINT AT B,C;                  CALL PRINT AT
74 LET A=8                        LD A,+N
75 PRINT CHR$ A                   RST 16
76 RETURN (da subrotina)          RET
```

A partir daqui podemos agora listar em formato assembler, e na totalidade, a primeira rotina em código máquina.

ETIQUETAS	MNEMÓNICAS	COMENTÁRIOS
TOPOS (RECTÂNGULO)	LDA, + 8	Gráfico 8
SUBROTINA 1	LDB, + 32	32 caracteres
	→ RST 16	ROTINA ROM (PRINT CHR$)
	DJNZ	Contador decrescente
	RET	RETORNO à Rotina principal
LADOS (RECTÂNGULO)	LDA, + 19	
SUBROTINA 2	SUB B	
	LDB,A	
	CALL PRINT AT	ROTINA ROM (PRINT AT)
	LDA, + 8	Gráfico 8
	RST 16	ROTINA ROM (PRINT CHR$)
	RET	RETORNO à Rotina principal
INÍCIO DA ROTINA	CALL SLOW	ROTINA ROM (SLOW)
	CALL CLS	ROTINA ROM (CLS)
	CALL (TOPOS)	SUBROTINA 1
	LDB, + 18	18 Linhas (1 a 18)
	→ PUSH BC	
	LDC, + 0	Coluna 0
	CALL (LADOS)	SUBROTINA 2
	POP BC	Ver resumo teórico
	PUSH BC	
	LDC, + 31	Coluna 31
	CALL (LADOS)	SUB ROTINA 2
	POP BC	
	DJNJ	Contador decrescente
	CALL (TOPOS)	SUBROTINA 1
	RET	RETORNO AO BASIC

Esta rotina em C.M., executa o mesmo trabalho que as linhas Basic (de 20 a 140) com uma diferença em tempo de aproximadamente 1 para 10.

Escreva:

10 REM XXXX (Reserve 53 caracteres).

Introduza os códigos listados em decimal, pelo processo habitual:

16514	62,	8,	6,	32,	215,	16,	253		
16521	201,	62,	19,	144,	71,	205,	245,	8	
16529	62,	8,	215,	201,	205,	43,	15		
16536	205,	42,	10,	205,	130,	64,	6,	18	
16544	197,	14,	0,	205,	138,	64,	193,	197	
16552	14,	31,	205,	138,	64,	193,	16,		
16559	240,	205,	130,	64,	201,	0,	0,	0,	0

NOTA: Termina no endereço 16567

Apague as linhas Basic até à 140 e acrescente:

20 LET K = USR 16533

Se tudo estiver certo, vai obter a formação do rectângulo do jogo, com uma rotina totalmente em código máquina.

(Cont. no próximo número)

ENCICLOPÉDIA DA LINGUAGEM BASIC

(Continuação do número anterior)

comando APPEND

Este comando combina um programa que está guardado numa zona externa, por exemplo numa cassette ou diskette, com outro programa que está presente na memória do computador.

O número de linha onde se inicia o segundo programa deve ser superior ao do programa que está na memória ou, de contrário, as novas linhas serão sobrepostas a todas aquelas que existiam e possuiam o mesmo número.

APPEND é usado frequentemente para ligar "files" muito extensas de DATA com programas que estão na memória.

Exemplos de utilização:

```
1000 PRINT "AS PRÓXIMAS LINHAS PERTENCEM
     AO PROGRAMA 2"
1010 APPEND "PROGRAMA 2"
1020 END
```

SE O SEU COMPUTADOR NÃO POSSUI ESTE COMANDO, VERIFIQUE O QUE SE PASSA COM OS COMANDOS:

MERGE ou WEAVE

QUE SÃO COMANDOS EQUIVALENTES.

NO CASO DO TRS80-MOD I, O procedimento é complicado! tomemos nota:

1 — Localizar o "ponteiro de memória" que indica o fim do programa (está guardado nas posições 16333 e 16334). Usar PEEK para obter estes valores.

2 — Subtrair 2 a partir do primeiro número (aquele que está na pos. 16333). No caso da diferença ser negativa, terá de adicionar + 256 e subtrair 1 do segundo número.

3 — Usar POKE para armazenar os dois números obtidos nas posições 16548 e 16549 sem outras modificações.

4 — Guardar o programa numa cassette com o comando CLOAD. Restaurar os valores primitivos em 16548/9 com POKEs de 233 em 16548 e 66 em 16549.

5 — Pode agora executar o APPEND do segundo programa.

SE A SUA MÁQUINA POSSUI O COMANDO MERGE, PODE DAR-SE POR FELIZ!

FUNÇÃO AT

A função AT é usada como instrução PRINT, de modo a especificar a posição onde o caracter ou a variável se deve situar.
Esta função pode ser um NÚMERO, uma VARIÁVEL NUMÉRICA, ou uma OPERAÇÃO MATEMÁTICA.
Uma vírgula (ou ponto e vírgula) deve ser inserida entre o valor de AT e a próxima cadeia de caracteres ou variável.

Por exemplo:

```
10 PRINT AT (2,12); "HOJE E DOMINGO"
20 PRINT AT (12,2); "AMANHÃ  TRABALHAMOS"
```

No caso das máquinas Sinclair, a primeira frase será escrita na linha 2, coluna 12. No caso, por exemplo, do TRS 80 mod I usaríamos PRINT AT 420, porque usa posições do ecran entre 0 e 1023.

Observe por exemplo:

```
10 INPUT X
20 PRINT AT 10 AND X>5,10; "X E MAIOR QUE 5"
30 GOTO 10
```

No caso de x ser superior a 5, verá que a string será escrita na linha 10 e se a condição for falsa, a string será escrita na linha zero.

SIMULAÇÃO

Última Parte/Jan. 84
Autor: ALEXANDRE SOUSA

Terminamos hoje o nosso estudo do programa de simulação que temos vindo a incluir desde o mês de Outubro.
Ao testarmos devidamente o programa, é natural acabarmos por alterar uma ou duas linhas das explicações anteriores.
Desta vez publicamos a listagem completa do programa. Chamamos a atenção para o facto de que as máquinas Sinclair não usam o índice zero nas variáveis indexadas, pelo que as linhas onde anteriormente usamos D (0) ou P (0) terão de ser modificadas.
Na linha inicial do programa, com a instrução RANDOMIZE, vai ser desencadeada a sucessão de números aleatórios.
Inicialmente damos entrada de um número entre 0 e 9 e a sequência da função RND vai variar de acordo com o número escolhido, que para nós é arbitrário.
As saídas que obtivemos em 10 vezes que fizémos correr o programa são as seguintes:

```
MEDIA-PREJUIZO DIARIO=[illegible]
MEDIA-RECEITA/DIARIA =[illegible]

MEDIA-PREJUIZO DIARIO=4874,8
MEDIA-RECEITA/DIARIA =5721,7

MEDIA-PREJUIZO DIARIO=[illegible]
MEDIA-RECEITA/DIARIA =[illegible]

MEDIA-PREJUIZO DIARIO=4905,8
MEDIA-RECEITA/DIARIA =5790,7

MEDIA-PREJUIZO DIARIO=4888,5
MEDIA-RECEITA/DIARIA =5727,3

MEDIA-PREJUIZO DIARIO=4939,7
MEDIA-RECEITA/DIARIA =5756,2

MEDIA-PREJUIZO DIARIO=4905,8
MEDIA-RECEITA/DIARIA =5790,7

MEDIA-PREJUIZO DIARIO=5077,7
MEDIA-RECEITA/DIARIA =5818,2

MEDIA-PREJUIZO DIARIO=[illegible]
MEDIA-RECEITA/DIARIA =[illegible]

MEDIA-PREJUIZO DIARIO=[illegible]
MEDIA-RECEITA/DIARIA =[illegible]
```

Podemos tomar as seguintes notas sobre a saída de dados:

— Todos os valores obtidos nas diferentes "corridas" do programa não são idênticos. Isso deve-se ao facto de o RANDOMIZE criar diferentes padrões do número de cliente à chegada.

— Existe uma pequena percentagem de variação entre os dados obtidos.

— Finalmente é de notar que existe uma perda significativa e que, provavelmente, valerá dinheiro a admissão de um novo empregado.

Este programa foi escrito em BASIC e aplicado a um exemplo concreto; no entanto, existem programas do tipo VISICALC que permitem a simulação financeira ou orçamental com grande simplicidade. No caso do ZX81/TMS 1000 e SPECTRUM, o programa denomina-se VUCALC e permite a elaboração de uma grelha de cálculo de 60 linhas por 60 colunas. Podemos definir fórmulas em quadrículas da grelha e, apenas com a variação de um parâmetro, obter os seus reflexos em todas as outras componentes.

```
   1 REM "SIMULACAO"
   2 INPUT "NUM.entre 0 e 2"
   3 RANDOMIZE
   4 LET C1=0: LET CL=0: LET L1=
0
  10 DIM A(18)
  11 DIM D(15)
  12 DIM P(15)
  20 DATA 30,15,6,3,3,25,9,6,12,
12,18,22
  30 FOR H=1 TO 12
  40 READ A(H+6)
  50 NEXT H
  60 PRINT "aguardar um momento"

 110 DATA 0,0,.15,.25,.15,.6,.35
,.3,.5,.2
 115 READ D(1),P(1)
 120 READ D(2),P(2)
 130 LET D(3)=D(2): LET D(4)=D(2)

 140 LET P(3)=P(2): LET P(4)=P(2)

 150 READ D(5),P(5)
 160 LET D(6)=D(5): LET D(7)=D(5)

 170 LET P(6)=P(5): LET P(7)=P(5)

 180 READ D(8),P(8)
 190 LET D(9)=D(8): LET D(10)=D(
8): LET D(11)=D(8)
 200 LET P(9)=P(8): LET P(10)=P(
8): LET P(11)=P(8)
 210 READ D(12),P(12)
 220 FOR J=12 TO 15
 230 LET D(J)=D(12): LET P(J)=P(
12)
 240 NEXT J
 250 FOR E=1 TO 10
 300 LET TH=7: LET TM=0
 310 LET L=1: LET PC=0: LET TV=0
 410 IF TM=0 THEN GO SUB 1200
 420 LET T=TM/4+1
 430 FOR J=1 TO D(T)
 440 LET C=INT (2*RND)+1
 450 IF C=1 THEN GO TO 500
 455 GO TO 600
 490 REM 500-550 SAIDA CLIENTE
 500 LET rd=RND
 502 IF rd>D(L) THEN GO TO 550
 510 LET PC=PC+57.50
 520 GO TO 690
 550 LET L=L+1: REM CLIENTE entr
a

 560 GO TO 690
 590 REM 600-660 CLIENTE FICA
 600 LET rd=RND
 602 IF rd>P(L) THEN GO TO 640
 610 LET PC=PC+57.50: REM (CLIEN
TE PERDIDO)
 620 GO TO 690
 640 LET L=L+1
 690 NEXT J
 700 IF L=1 THEN GO TO 710
 702 LET L=L-1
 704 LET TV=TV+57.50: REM (CLIEN
TE GANHO)
 710 GO SUB 1000: REM (ACTUALIZA
R O TEMPO)
 720 IF TH=19 THEN GO TO 1500
 725 GO TO 800
 730 REM TH=19: REM (FINAL DO DI
A)

 800 GO TO 410: REM (PROXIMO SEG
MENTO DE 4 MINUTOS)
1000 REM AVANCO DO TEMPO
1010 LET TM=TM+4
1020 IF TM>=60 THEN GO TO 1030
1025 GO TO 1100
1030 LET TM=TM-60
1040 LET TH=TH+1
1100 RETURN
1200 FOR S=1 TO 15
1210 LET D(S)=0
1220 NEXT S
1230 FOR I=1 TO A(TH)
1240 LET X=INT (15*RND)+1
1250 LET D(X)=D(X)+1
1260 NEXT I
1270 RETURN
1500 PRINT "FINAL DO-";E;"- DIA
ESTATISTICO"
1510 PRINT "____________________
________"

1515 LET L1=PC+L1+L*57.50
1518 LET C1=TV+C1
1520 LET CL=L+CL
1700 NEXT E
1800 LET L1=INT (L1+5)/10
1810 LET C1=INT (C1+5)/10
1820 LET CL=INT ((CL/10)+.5)
1825 CLS
1830 PRINT "MEDIA-PREJUIZO DIARI
O-";L1

1840 PRINT "MEDIA-RECEITA DIARIA
-";C1
1850 PRINT "Med.-Num.C.../HOR.FEC
HO -";CL
```

QUICK LOAD

ZX81 16K

In. YOUR COMPUTER, Junho/83
Adapt. de ROCHA BARBOSA/V. N. Gaia

Este programa em código máquina permite-lhe carregar, gravar e verificar * programas em, pelo menos, o dobro da velocidade normal e, provavelmente, com mais segurança.
O código máquina fica na "REM statement" no início do programa. Escreva:

1 REM (256Xs)

Para verificar se cometeu algum erro na entrada das 256 letras "X" faça

PRINT PEEK 16770

o que deverá dar 118.
Agora, o código hexadecimal da listagem 1 é carregado cuidadosamente através do programa 2, podendo dar entrada de mais que um byte de cada vez.
Por exemplo:
CDE702 em vez de CD.E7.02 (o ponto representa NEWLINE).

Quando terminar de dar entrada de todo o código faça NEWLINE e a máquina parará com um erro. Não se preocupe. Faça:

POKE 16510,0

Este "POKE" transformará a linha 1 na linha 0, protegendo o código máquina de qualquer erro. Seria bom até fazer duas cópias numa cassete prevendo que o programa possa "ir ao ar".
Para se certificar de que tudo está correcto, faça RAND USR 16514 (não faça NEWLINE).
Pegue numa cassete, preferivelmente de boa qualidade, ligue o gravador para gravação e agora NEWLINE. O programa será gravado normalmente, excepto as linhas do ecran que devem ser mais juntas e numerosas. Se isto não se verificar é porque cometeu algum erro na introdução do código máquina.
Recarregue uma cópia do programa que deveria ter feito previamente e dê entrada da listagem do programa 3. Depois de fazer RUN, pode corrigir os erros detectados e gravar de novo.
Para verificar a gravação, procure na fita o início do programa e faça:

RAND USR 166601 (sem NEWLINE).

Inicie o gravador e NEWLINE. A verificação do programa é feita normalmente excepto, como já referimos, as linhas que aparecerão mais juntas.

Quando completa esta operação, a máquina deve parar com a mensagem 0/0. No entanto, pode parar com a mensagem de erro R/0, o que significa que o conteúdo da gravação não coincide com o programa em memória; pode ter havido erros ao nível do volume (é necessário, por vezes, um nível mais alto que o de carregamento normal). Altere o volume e tente de novo. Se ainda não funcionar, talvez esteja a haver interferência de uma fonte exterior; verifique se os cabos de alimentação não estão cruzads. Faça uma nova gravação e tente verificá-la de novo. Confirme também se não houve incorrecções na entrada do programa.
Se a verificação se deu normalmente tire todas as linhas, excepto a que contém o código máquina, como é lógico.
Vamos agora realizar o "LOAD" de um programa. Faça:

RAND USR 16607 (sem NEWLINE).

Procure na cassete o início da gravação do programa, inicie o gravador e NEWLINE. Deve terminar com a mensagem 0/0. Se assim não acontecer, tente uma nova cópia porque se o programa foi verificado sem erro também será carregado de igual modo.
Se decorreu sem erros faça uma cópia normal do programa (retire todas as linhas, excepto a que contém o código máquina — antes da gravação).

As principais funções do QUICK LOAD:

SAVE	RAND USR 16514	(gravação do programa e variáveis).
VERIFY	RAND USR 16601	(verifica se o conteúdo da fita é igual ao que se encontra em memória)
LOAD	RAND USR 16607	(limpa o programa anterior e as variáveis, e carrega o novo programa).

Se fizer BREAK durante qualquer destas operações, a máquina parará com um erro "D".
Os comandos podem ser usados em linhas do programa ou dando entrada directa deles. Se o comando SAVE estiver no programa, então a execução da linha do próximo programa ocorrerá depois deste ter sido carrregado.
Para carregar um programa em alta velocidade, deve estar incluído o QUIK LOAD e os programas que

deseja gravar em alta velocidade não devem conter código máquina. Siga este método:

1.º — Baixe a RAMTOP dando entrada de POKE 16389,127 e NEWLINE.
2.º — Carregue o QUICK LOAD.
3.º — Acrescente a listagem 4 e faça RUN em modo FAST; NEWLINE e carregue o programa em que pretende usar o QUICK LOAD "à cabeça".
4.º — Faça uma linha 1 (e não 0) REM (256 Xs) e as restantes linhas da listagem 5 (10, 20, 30, 40,); RUN em modo FAST.
5.º — Faça: POKE 16510,0 — para proteger a linha REM, passando a ser linha 0.
6.º — Grave e verifique o programa em alta velocidade.

Note, que devido à listagem do programa 5, pode por vezes acontecer ter que alterar a listagem do programa onde quer utilizar o QUICK LOAD, de modo a que as linhas não coincidam.
Sugerimos que utilize o QUICK LOAD da seguinte forma:
Grave de um lado da cassete os programas gravados com o QUICK LOAD. Grave uma cópia do Q.L. à velocidade normal. Quando pretender carregar um programa dessa cassete, deve primeiro meter o Q.L. e usar o seu comando (Q.L.) "LOAD".

NOTAS:

— Verificou-se que há pelo menos uma possibilidade de dar nome a um programa. Quando o programa a gravar está no computador e o Q.L. se encontra na linha 0, grave e verifique como já explicámos. Carregue o Q.L. e junte as linhas:

```
1 SAVE "nome"
2 RAND USR 16607
```

Introduza o programa e faça RUN. Repare que o computador fica à espera da entrada de um programa. Desligue o gravador e faça BREAK. Pegue na cassete que contém o programa a utilizar e carregue-o em alta velocidade com RAND USR 16607.
Agora grave o programa em alta velocidade fazendo RAND USR 16514 (sem NEWLINE).
Inicie o gravador para gravar e NEWLINE.
Quando quiser fazer LOAD deste programa utilize:

```
LOAD "nome"
```

— Pode também utilizar o QUICK LOAD em programas com código máquina. Apenas terá que alterar a linha do programa onde se encontra o endereço RAND USR "endereço" para (endereço) + 262.

ATENÇÃO — Repare que não pode ter "JUMPS INCONDICIONAIS" ("saltos" para o endereço do Q.L.).

* Verificar é uma forma de se assegurar que um programa foi gravado nas melhores condições.

LISTAGEM 1

```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
16586 - CE C3 07 02
16590 - 3E FF 32 27
16594 - 40 FD CB 3B
16598 - 3B CF 0C FD
16602 - CB 09 CE 16
16606 - 04 FD CB 09
16610 - 3E 1C E7 0E
16614 - FD 1C 09 3E
16618 - 18 1C 0E 01
16622 - 06 00 3E 7F
16626 - DB FE D3 FF
16630 - 1F 30 47 17
16634 - 17 38 57 10
16638 - F1 F1 FD CB
16642 - 00 46 20 40
16646 - 21 0A 40 11
16650 - 0F 41 D5 18
16654 - DD FD CB 00
16658 - D5 FD CB 00
16662 - 4E 23 04 73
16666 - BE 20 3B 71
16670 - 23 EB 2A 14
16674 - 40 37 ED 52
16678 - EB 30 E0 22
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible] - 5A [illegible]
[illegible]
```

PROGRAMA 2

```
10 LET A=16514
20 LET A$=""
30 IF A$="" THEN INPUT A$
40 POKE A,16*CODE A$+CODE A$(2
   -476
50 SCROLL
60 PRINT A,A$ (TO 2)
70 LET A$=A$(3 TO)
80 LET A=A+1
90 GOTO 30
```

PROGRAMA 3

```
1000 LET A=16514
1010 PRINT A;" - "
1020 FOR B=A TO A+7
1030 LET C= PEEK B
1040 PRINT CHR$(28+INT(C/16));
     CHR$(28+(C-16*INT(C/16)));
     " ";
1050 NEXT B
1060 LET A=B
1070 GOTO 1010
```

PROGRAMA 4

```
10 FOR A=0 TO 250
20 POKE 32512+A,PEEK (16514+A)
30 NEXT A
```

PROGRAMA 5

```
0  REM (256 X'S)
10 FOR A=0 TO 250
20 POKE 16514+A,PEEK(32512+A)
30 NEXT A
40 STOP
```

ANAGRAMA SPECTRUM/Zx81/TMS 1000

CARLOS SILVA
Tomar

Este programa escreve todas as possibilidades de combinações existentes entre 5 caracteres.

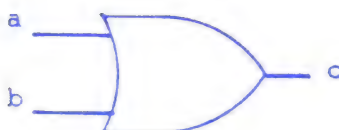
```
  5 REM "ANAGRAMA"
 10 DIM A$(5,1)
 20 FOR F=1 TO 5
 30 INPUT A$(F)
 32 PRINT A$(F)
 35 NEXT F
 40 PRINT
 60 FOR A=1 TO 5
 70 FOR B=1 TO 5
 80 IF B=A THEN GO TO 310
 90 FOR C=1 TO 5
100 IF C=A OR C=B THEN GO TO 30
0
110 FOR D=1 TO 5
120 IF D=A OR D=B OR D=C THEN G
O TO 290
130 LET E=15-(A+B+C+D)
200 PRINT A$(A);A$(B);A$(C);A$(
D);A$(E),
290 NEXT D
300 NEXT C
310 NEXT B
320 NEXT A
```

SIMULAÇÃO DE CIRCUITOS LÓGICOS SPECTRUM/ZX81/TS 1000

Autor: MALCOM FARNSWORTH
Adapt. de ALEXANDRE SOUSA

PARTE II

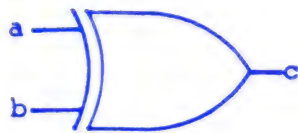
O R

a	b	c
0	1	1
1	0	1
0	0	0
1	1	1

```
1)LET A(C,T)=1

2)IF A(A,T)=0 AND A(B,T)=0 THEN
LET A(C,T)=0
```

N O R

a	b	c
0	1	0
1	0	0
0	0	1
1	1	0

```
1)LET A(C,T)=0

2)IF A(A,T)=0 AND A(B,T)=0 THEN
LET A(C,T)=1
```

O R -EXCLUSIVO

a	b	c
0	1	1
1	0	1
0	0	0
1	1	0

```
1)LET A(C,T)=1
2)IF A(A,T) = A(B,T) THEN LET A(C
,T)=0
```

FLIP-FLOP tipo J-K

f	a	b	c	d	e
0	x	x	x	0	1
1	1	x	x	0	0
1	-	0	0	0	0
1	-	1	0	1	0
1	-	1	0	0	1
1	-	1	1	C	C

observacoes

x qualquer valor
- descida do impulso(clock)
0 a saida fica invariavel
C saida comutada

```
1) LET A(D,T)=A(D,T-1)
2) IF A(A,T-1)=1 AND A(A,T)=0
AND A(B,T-1)=1 AND A(C,T-1)=1
THEN LET A(D,T)=1-A(D,T-1)
3) IF A(A,T-1)=1 AND A(A,T)=0
AND A(B,T-1)=1-A(C,T-1) THEN
LET A(D,T)=A(B,T-1)
4) IF A(F,T)=0 THEN LET A(D,T)=0
5) LET A(E,T)=1-A(D,T)
```

I N V E R S O R

A	B
0	1
1	0

```
LET A(B,T)=1-A(A,T)
```

FLIP-FLOP tipo D

e	f	a	b	c	d
0	1	x	x	1	0
[illegible]	0	x	x	[illegible]	1
[illegible]	0	x	x	[illegible]	1
1	1	+	1	1	0
1	1	+	0	0	1
1	1	0	x	0	0

observacoes

* estado instavel
+ subida do impulso(clock)
: qualquer valor
0 a saida fica invariavel

```
1) LET A(C,T)=A(C,T-1)
2) IF A(A,T-1)=0 AND A(A,T)=1
THEN LET A(C,T)=A(B,T-1)
3) IF A(F,T)=0 THEN LET A(C,T)=0
4) IF A(E,T)=0 THEN LET A(C,T)=1
5) IF A(E,T)=0 AND A(F,T)=0 THEN
 PRINT "ERRO"
6)LET A(C,T)=1-A(C,T)
```

```
   1 REM PROGRAMA ADAPTADO E TRA
DUZIDO POR JOAQUIM MAGALHAES (OG
/PORTO
   2 REM pode ser usado com SPEC
TRUM-ZX81-TS1000
  10 PRINT "N. DE NOS NO CIRCUIT
O? (2-OU >2 )"
  20 INPUT N
  30 IF N>=2 THEN GO TO 50
  40 GO TO 20
  50 PRINT N
  60 PRINT
  70 PRINT "N. DE IMPULSOS? (2-2
9)"
  80 INPUT T
  90 IF T>=2 AND T<=29 THEN GO T
O 110
 100 GO TO 80
 110 PRINT T
 120 FOR X=1 TO 70
 130 NEXT X
 150 CLS
 510 DIM A(N,T)
 520 FOR X=1 TO N
 530 FOR Y=1 TO T
 540 LET A(X,Y)=0
 550 NEXT Y
 560 NEXT X
 810 FOR Y=2 TO T STEP 2
 820 LET A(1,Y)=1
 830 NEXT Y
1010 FOR Y=3 TO 6
1020 LET A(2,Y)=1
1030 NEXT Y
1510 LET A(6,1)=1
2010 FOR Y=2 TO T
2020 LET A(3,Y)=a(3,Y-1)
2030 IF A(1,Y-1)=0 AND A(1,Y)=1
```

```
THEN LET A(3,Y)=A(2,Y-1)
2040 LET A(4,Y)=A(4,Y-1)
2050 IF A(1,Y-1)=0 AND A(2,Y-1)=1
THEN LET A(4,Y)=A(3,Y-1)
2060 LET A(5,Y)=A(5,Y-1)
2070 IF A(1,Y-1)=1 AND A(2,Y-1)=2
AND A(3,Y-1)=1 AND A(4,Y-1)=1 TH
EN LET A(5,Y)=1-A(5,Y-1)
2080 IF A(1,Y-1)=1 AND A(2,Y-1)=2
AND A(3,Y-1)=2-A(4,Y-1) THEN LET
 A(5,Y)=A(2,Y-1)
2090 LET A(6,Y)=1
2100 IF A(5,Y)=1 AND A(4,Y)=1 TH
EN LET A(6,Y)=0
2110 NEXT Y
4010 PRINT "1 FORMA DE ONDA"
4020 PRINT "2 TABELA"
4030 PRINT "3 CIRCUITO"
4040 PRINT
4050 PRINT "1,2 OU 3 ""ENTER"""
4060 INPUT B
4065 CLS
4070 IF B=1 THEN GO TO 5000
4080 IF B=2 THEN GO TO 6000
4090 IF B=3 THEN GO TO 7000
4100 GO TO 4010
5010 LET A$="N T12345678901234
7890123456789"
5020 PRINT A$(1 TO T+3)
5030 FOR X=1 TO N
5040 PRINT X;TAB 3;
5050 FOR Y=1 TO T
5060 LET P=A(X,Y)
5070 IF P=0 THEN PRINT "[illegible]"
5080 IF P=1 THEN PRINT "[illegible]"
5090 NEXT Y
5100 PRINT
5110 PRINT
5120 NEXT X
5130 STOP
6000 PRINT "[illegible]"
6010 LET A$="_ _12345678901234
7890123456789"
6012 LET B$="____________________"
6020 PRINT A$(1 TO N+3)
6022 PRINT B$(1 TO N+3)
6030 FOR Y=1 TO T
6040 PRINT Y;TAB 3;
6050 FOR X=1 TO N
6060 PRINT A(X,Y);
6070 NEXT X
6080 PRINT
6090 NEXT Y
6100 STOP
7010 LET Y=15
7020 LET X=2
7030 GO SUB 7500
7040 LET X=9
7050 GO SUB 7500
7060 LET Y=15
7070 LET X=13
7080 GO SUB 7600
7090 LET X=23
7100 GO SUB 7700
7110 PRINT AT 16,2;"1"
7120 PRINT AT 18,0;"3"; "STOP"
7130 STOP
7510 PRINT AT [illegible]
7520 PRINT AT [illegible]
7530 PRINT AT [illegible]
7540 PRINT AT [illegible]
7550 PRINT AT [illegible]
7560 RETURN
7600 PRINT AT [illegible]
7610 PRINT AT [illegible]
7620 PRINT AT [illegible]
7630 PRINT AT [illegible]
7640 PRINT AT [illegible]
7650 RETURN
7700 PRINT AT [illegible]
7710 PRINT AT [illegible]
7720 PRINT AT [illegible]
7730 PRINT AT [illegible]
7740 PRINT AT [illegible]
7750 RETURN
```

(FIM)

ANÁLISE ESTATÍSTICA

SPECTRUM 48K

In. ZX COMPUTING, N.º 9 — Vol. 1
Adapt. de J. MAGALHÃES / Clube Z80

Este programa está testado e preparado para o SPECTRUM 48K, mas pode ser adaptado para 16K ou para o ZX81/TMS 1000 ou para qualquer outro tipo de BASIC. Para converter este programa para outra máquina, tem de ter em atenção as instruções com POKE-BEEP-INK-OVER.

Trata-se de um programa razoável capaz de efectuar muitos dos testes estatísticos considerados importantes e que possibilitam analisar grande quantidade de dados.

Entre as diferentes opções encontram-se o DESVIO PADRÃO, COEFICIENTE DE VARIAÇÃO, ERRO STANDARD DA MÉDIA, TESTE TIPO F, MODA, REGRESSÃO, COEFICIENTE DE CORRELAÇÃO, TESTE TIPO T STUDENT E TESTE DE SIGNIFICÂNCIA. Há ainda a possibilidade de efectuar o gráfico da sua análise e copiar os resultados na sua impressora. Também está incluída uma rotina para obter o gráfico de barras — ela é complexa e determina os valores máximo e mínimo de modo a calcular a gama de valores, ordenar os dados por grupos e imprimir os resultados no ecran.

Qualquer página de resultados pode ser copiada para a impressora, usando a tecla "C". Todavia, se usar longas listas de dados, é preferível fazer uma conversão para LPRINT.

Mantemos nesta adaptação a rotina original em BASIC que executa a ordenação — 15 minutos para 100 pares de valores. Francamente parece-nos um exagero, mas prometemos publicar outra rotina que melhore significativamente esta zona do programa.

INSTRUÇÕES

Este programa de estatística descritiva executa cálculos de modo a fornecer o valor da MÉDIA de dois conjuntos de dados. Dá também a medida da distribuição de valores próximos da média, que são:

a) DESVIO PADRÃO — pode ser um valor positivo (+) ou negativo (−). A 95% do limite de confidên-

cia, 95,4% de todos os valores deverão estar dentro dos limites +2 e −2, a partir da média. Qualquer valor fora destes limites é considerado fora da média dos dados normalmente distribuídos.

b) COEFICIENTE DE VARIAÇÃO (CV) — trata-se de uma percentagem que se usa para representar a precisão dos dados. Relaciona o DESVIO PADRÃO e a MÉDIA de forma que CV=DP/MD * 100%. Um valor baixo indica elevada precisão.

c) ERRO PADRÃO (DA MÉDIA) — indica a proximidade do valor obtido para a média da amostragem, em relação à média da população; este valor pode não ser relevante para o conjunto de dados. EPM =DP/SQR(N) de modo que, ao aumentar o número de dados, este valor diminui e aproxima-se da média "verdadeira" da população (em que EPM=0).

F-TESTE

Este teste é usado algumas vezes para verificar a precisão de dois grupos de dados.

O valor F é calculado a partir da razão entre Variâncias (DP ao quadrado) e pode ser comparado em tabelas, a partir das quais é também determinado o valor da probabilidade (P). Isto dá acesso a significância (se existir) entre dois valores da precisão (ver também as notas para o T-TESTE).

A segunda escolha do Menu, permite a listagem dos dados. Os mesmos podem ser copiados, usando a tecla "C"em vez de "Enter".

MEDIANA E ORDENAÇÃO

A terceira opção e parte final da Estatística Descritiva usa uma rotina de ordenação no sentido descendente que, uma vez executada, permite ver o valor máximo mediana e valor mínimo para X e Y. A MEDIANA é o valor equivalente à divisão do número de observações em duas partes iguais. Em dados normalmente distribuídos, a mediana deve ser muito próxima do valor da média.

As diferenças entre a média e a mediana mostram-nos o deslizar (SKEW) para a esquerda ou direita.

A MODA é um valor que se determina com facilidade em conjuntos de dados normalmente distribuidos. A MODA, a MÉDIA e a MEDIANA devem ser (aproximadamente) iguais.

Os SKEWs para a esquerda e direita devem ser significativamente diferente um do outro.

ANÁLISE ESTATÍSTICA

Os cálculos que podemos obter são:

REGRESSÃO — CORRELAÇÃO — T-TESTE

REGRESSÃO — trata-se de uma estimativa da associação entre os dados tipo X e os dados tipo Y.

Existe REGRESSÃO LINEAR quando essa associação é representada como uma linha recta, tal que y=b*x+c, em que c intercepta o eixo dos XX e b (coeficiente da regressão) é o gradiente.

Se a associação é totalmente linear, então b será =+1 ou −1, dependente da inclinação desta recta.

Estatisticamente, a regressão é usada para traçar a linha melhor adaptada aos dados.

b é calculado para determinar o valor de c (c=C−b*B). A linha é desenhada a partir de c e através de B e C até ao limite superior dos dados.

Este método é usado na parte gráfica e só trabalha para correlações positivas. Se a correlação fosse negativa, então a linha sairia fora do ecran.

Este ponto pode ser melhorado (linhas 1750; 1760).

COEFICIENTE DE CORRELAÇÃO (r)

Trata-se da expressão que associa dois conjuntos de dados. A correlação completa será =+1 ou −1, dependendo da inclinação da linha.

Se os dados não forem correlacionados, o valor será r=0.

Correlação e Análise de Regressão são frequentemente tratados em conjunto e fazem parte dos valores saídos no ecran.

AMOSTRAGEM DO T-TESTE

Existem vários tipos de t-teste. O que vamos usar é para duas amostragens de igual comprimento. O teste é usado para determinarmos onde existe diferença significativa entre dois conjuntos de dados.

O valor t é calculado com o número de GRAUS DE LIBERDADE (GL). A partir das tabelas t-Tables pode determinar-se o valor probabilístico (p) para teste de significância.

O valor (p) pode ser expresso como A% para ver se existe inferência estatística entre dois conjuntos de dados — por exemplo p<=0.001 ou 0.1% significa que os dados são estatisticamente os mesmo. Se o valor t excede os valores tabelados, então a diferença é significativa.

GRÁFICO LINEAR

Está limitado pelo ECRAN e permite apenas valores para (x,y) limitados a (26,20), os médios inferiores a 11. Também é de notar que assume correlação positiva linear, entre x e y, quando desenha uma linha entre os pontos dados. No entanto, é de prever que tenham de ser eliminados alguns pontos que alteram o traçado.

Se é necessário o gráfico de valores elevados, terá de dar entrada desses valores, afectada de uma redução à escala pretendida.

A rotina da parte gráfica permite dar nomes aos eixos respectivos e especifica o comprimento de cada eixo.

```
 TO 250
 590 IF q$="n" OR q$="N" THEN GO
 TO 700
 595 GO TO 570
 600 REM GRAFICOS
 610 DATA 0,126,66,32,24,32,66,1
26
 620 DATA 255,0,195,102,60,60,10
2,195
 630 DATA 255,0,195,102,60,24,48
,224
 650 REM IMPRESSAO
 660 INPUT "ENTER continuar- C i
mprimir ";w$
 670 IF w$="C" THEN COPY : CLS :
 RETURN
 680 IF w$="" THEN CLS : RETURN
 690 GO TO 650
 700 PRINT PAPER 1; INK 6; FLASH
 1;"Espere um momento."''': PRIN
T PAPER 6; INK 1; FLASH 0;"ESTOU
 A CALCULAR OS VALORES     ESTAT
ISTICOS."
 710 GO SUB 7000: CLS
 720 REM MENU
 730 CLS : BEEP .8,30: PRINT INK
 1;"Qual a sua opcao ?"''
 740 PRINT INK 2;"A- Estatistica
 Descritiva:"; OVER 1;AT 3,3;"
                  "'': PRINT I
NK 1;"""1"" MEDIA,DP,CV(%),EPM,
 f-TESTE": PRINT """2"" IMPRIMIR
 DADOS": PRINT """3"" MIN, MAX &
 MEDIANA"''
 750 PRINT INK 2;"B- Analise Est
atistica:"; OVER 1;AT 9,3;"
          "'': PRINT INK 1;"
""4"" REGRESSAO E CORRELACAO": P
RINT """5"" AMOSTRAGEM DO t-TEST
E"''
 760 PRINT INK 2;"C- Graficos:";
 OVER 1;AT 14,3;"        "'': PRINT
 INK 1;"""6"" GRAFICO LINEAR": P
RINT """7"" GRAFICO DE BARRAS"''
: PRINT INK 3;"""0"" SAIDA DO PR
OGRAMA"
 770 INPUT w$: IF w$="" THEN GO
TO 720
 780 LET w=VAL w$: CLS
 790 IF w=0 THEN GO TO 6000
 800 IF w>0 AND w<4 THEN GO TO 8
00+w*100
 810 IF w>4 AND w<7 THEN GO TO 1
000+w*100
 820 IF w=4 THEN GO TO 1300
 830 IF w=7 THEN GO TO 2000
 840 IF w>7 THEN GO TO 720
 900 REM ESTATISTICA DESCRITIVA
 910 REM MEDIA, DP, CV(%), EPM,
f-TEST
 915 BEEP .8,30: PRINT INK 3;"ES
TATISTICA DESCRITIVA"; OVER 1;A
T 0,0;"                       "''
 920 LET s3=INT (s3*10↑5+.5)/(10
↑5): LET s4=INT (s4*10↑5+.5)/(10
↑5)
 930 LET v1=INT (v1*10↑3+.5)/(10
↑3): LET v2=INT (v2*10↑3+.5)/(10
↑3)
 940 LET s5=INT (s5*10↑5+.5)/(10
↑5): LET s6=INT (s6*10↑5+.5)/(10
↑5)
 950 LET h1=s3↑2: LET h2=s4↑2
 955 IF h1>h2 OR h1=h2 THEN LET
h3=h1/h2
 960 IF h2>h1 THEN LET h3=h2/h1
 965 LET h3=INT (h3*10↑4+.5)/(10
↑4): LET h4=a-1
 970 PRINT "N= ";a;"'","(b,c)= ";(
b,c)... PRINT "D.P. de x=
";s3: PRINT "C.V.(%) de x= ";v1:
PRINT "E.P.M. para x=";s5
 975 PRINT "D.P. de y=";s4: PRIN
T "C.V.(%) de y=";v2: PRINT "E.P
.M. para y=";s6''
 980 PRINT INK 3;"f-TEST:"; OVER
 1;AT 12,0;"       "'': PRINT INK
 1;"Teste para comparar a precis
ao de x com y.": PRINT "O valor
de f=";h3: PRINT "Para G.L.1 & G
.L.2 = ";h4: PRINT "Este valor p
ode ser procurado   nas f-TABELA
S para encontrar a  probabilidad
e de um valor (P)   para teste d
e significancia."
 990 GO SUB 650: GO TO 720
1000 REM IMPRIMIR
1010 BEEP .8,30: PRINT INK 2;"DA
DOS:"; OVER 1;AT 0,0;"     "''
1020 IF a>21 THEN GO TO 1060
1030 LPRINT "DADOS:"; OVER 1;AT
0,0;"      "'': FOR n=1 TO a
1040 LPRINT "n: ";n;"   ";"x: ";x(n
),"y: ";y(n)
1050 NEXT n: GO TO 1090
1060 FOR n=1 TO a
1070 PRINT "n: ";n;"   ";"x: ";x(n)
,"y: ";y(n)
1080 NEXT n
1090 PRINT : GO SUB 650: GO TO 7
20
1100 REM
1110 PRINT PAPER 1; INK 7; FLASH
 1;"Espere um momento."''': PRIN
T PAPER 6;: INK 1; FLASH 0;"ESTO
U A ORDENAR OS DADOS"
1111 REM VALORES DE X
1112 LET mm=a
1113 LET mm=INT (mm/2)
1114 IF mm=0 THEN GO TO 1130
1115 LET kk=a-mm
1116 LET jj=1
1117 LET ii=jj
1118 LET ll=ii+mm
1119 IF b(ii)<=b(ll) THEN GO TO
1124
1120 LET tt=b(ll): LET b(ll)=b(i
i): LET b(ii)=tt
1121 LET ii=ii-mm
1122 IF ii<1 THEN GO TO 1124
1123 IF NOT ii<1 THEN GO TO 1118
1124 LET jj=jj+1
1125 IF jj>kk THEN GO TO 1113
1126 GO TO 1117
1130 REM
1131 LET mm=a
1132 LET mm=INT (mm/2)
1133 IF mm=0 THEN GO TO 1160
1134 LET kk=a-mm
1135 LET jj=1
1136 LET ii=jj
1137 LET ll=ii+mm
1138 IF c(ii)<=c(ll) THEN GO TO
1143
1139 LET tt=c(ll): LET c(ll)=c(i
i): LET c(ii)=tt
1140 LET ii=ii-mm
1141 IF ii<1 THEN GO TO 1143
1142 IF NOT ii<1 THEN GO TO 1137
1143 LET jj=jj+1
1144 IF jj>kk THEN GO TO 1132
1145 GO TO 1136
1150 REM IMPRESSAO DOS VALORES
1160 CLS : BEEP .8,30: PRINT INK
 3;"DADOS ORDENADOS:"; OVER 1;AT
 0,0;"                "''
1165 FOR l=1 TO a
1170 PRINT "n: ";l;"   ";"x: ";b(l)
,"y: ";c(l)
1180 NEXT l
1190 PRINT : GO SUB 650
1200 REM MIN, MAX & MEDIANAS
```

```
1205 PRINT INK 3;"MIN, MAX, MEDI
ANAS.": OVER 1;AT 0,0;"__________
_______"
1210 LET s=a/2: IF s=INT s THEN
GO TO 1230
1220 LET m1=b(s): LET m2=c(s): G
O TO 1240
1230 LET w=a/2: LET v=a/2+1: LET
 m1=(b(w)+b(v))/2: LET m2=(c(w)+
c(v))/2
1240 PRINT "MEDIANA de x=";m1: P
RINT "O VALOR MINIMO DE x=";b(a)
: PRINT "O VALOR MAXIMO DE x=";b
(1)''
1250 PRINT "MEDIANA de y=";m2: P
RINT "O VALOR MINIMO DE y=";c(a)
: PRINT "O VALOR MAXIMO DE y=";c
(1)
1260 LET m1=INT (m1*10+.5)/10: I
F m1=INT (p*10+.5)/10 OR m1+.1=I
NT (p*10+.5)/10 OR m1-.1=INT (p*
10+.5)/10 THEN PRINT ''"MEDIANA
de x aprox' igual a    MEDIA de
 x indicando se os dadosestao no
rmalmente distribuidos."''.
1270 LET m2=INT (m2*10+.5)/10: I
F m2=INT (q*10+.5)/10 OR m2+.1=I
NT (q*10+.5)/10 OR m2-.1=INT (q*
10+.5)/10 THEN PRINT "MEDIANA de
 y aprox' igual a    MEDIA de y
 indicando se os dadosestao norm
almenta distribuidos."
1280 GO SUB 650: GO TO 720
1300 REM INICIO DA ANALISE ESTAT
ISTICA
1310 REM REGRESSAO E CORRELACAO
1320 BEEP .8,30: PRINT INK 3;"RE
GRESSAO E CORRELACAO:"; OVER 1;A
T 0,0;"
   ...."

1330 LET e=INT (e*10↑5+.5)/(10↑5
): LET d=INT (d*10↑5+.5)/(10↑5):
 LET u=INT (u*10↑5+.5)/(10↑5)
1340 PRINT "ax=";t1,"ax*x=";t3,"
ay=";t2,"ay*y=";t4,"ax*y=";t5
1350 PRINT "n=";a,"","(b,c)=";"(
";p;",";q;")"''': PRINT PAPER 1;
INK 6;"Para y=b*x+c"''': PRINT "b
=";d;"","c=";e'': PRINT INK 3;"C
OEF. CORREL. r=";u''
1360 IF d>.75 AND u>.75 THEN PRI
NT "b e r sao >.75 indicando uma
 forte correlacao linear positiv
a entre x e y."''
1370 IF d<-.75 AND u<-.75 THEN P
RINT "b e r sao <.75 indicando u
ma forte correlacao linear negat
iva entre x e y."''
1375 IF d<.75 AND u>.75 THEN PRI
NT "Coeficiente de correlacao (r
)<.75 indicando um forte relacio
namento positivo entre x e y."
1380 IF u=0 THEN PRINT "r=0 indi
ca que x e y sao totalmente nao
correlacionados."
1390 GO SUB 650: GO TO 720
1500 REM t-TEST e GL
1510 BEEP .8,30: PRINT INK 3;"2-
 AMOSTRAGEM  t-TESTE:"; OVER 1;A
T 0,0;"
   ...."
1515 LET t=INT (t*10↑5+.5)/(10↑5
)
1520 PRINT PAPER 4; INK 1;"Para
GL=";u2''
1530 PRINT INK 2;"O valor de t="
;t''
1540 PRINT INK 1;"Procure os val
ores de t nas tabelas t-TABLES a
 cima de GL e encontre os valore
s provaveis de (p) para um teste
 significativo."''
1550 GO SUB 650: GO TO 720
1600 REM SECCAO DE GRAFICOS
1605 REM LINHA
1610 BEEP .8,32: PRINT INK 3;"Li
nha de x,y:"; OVER 1;AT 0,0;"
   ...."

1620 PRINT INK 1;"SEGUIU A OPCAO
 ""3""              (Rotina das
medianas) ? (s/n):": INPUT i$: I
F i$="n" OR i$="N" THEN GO TO 72
0
1630 IF NOT i$="s" OR i$="S" THE
N CLS : GO TO 1600
1640 PRINT "Nome p/ Eixo de x":
INPUT a$
1650 PRINT "Nome p/ Eixo de y":
INPUT b$: CLS
1660 LET f=b(1): LET g=c(1)
1670 IF f>26 THEN LET f=26
1680 IF g>18 THEN LET g=18
1690 FOR o=17 TO (8*f+40): PLOT
o,9: NEXT o
1700 FOR m=9 TO (8*g+22): PLOT 1
7,m: NEXT m
1710 PRINT AT 21,3;a$: PRINT AT
18-g,2;b$: PRINT AT 19-g,0;g: PR
INT AT 21,2.5+f;f: PRINT AT 21,1
;"0"
1720 FOR n=1 TO a
1730 IF x(n)>=26 OR y(n)>=18 THE
N PRINT AT 19,3;"N. de dados exa
gerado": GO SUB 650: GO TO 720
1735 BEEP .02,30: INK 2: OVER 1:
 CIRCLE 8*x(n)+17,8*y(n)+9,2
1740 NEXT n
1750 IF p>10 OR q>10 THEN PRINT
AT 19,3;"Valor da MEDIA exagerad
o": GO SUB 650: GO TO 720
1760 FOR z=1 TO 20: BEEP .05,50-
z: NEXT z: INK 1: PLOT 17,8*e+9:
 DRAW p+16*p,q+16*p: PAUSE 200
1770 GO SUB 650: GO TO 720
2000 REM GRAFICO DE BARRAS
2010 BEEP .8,30: PRINT INK 3;"GR
AFICO DE BARRAS"; OVER 1;AT 0,0;
"
_______________________"
2020 PRINT INK 1;"SEGUIU A OPCAO
 ""3""              (Rotina das
Medianas)? (s/n)    Isto porque
sao necessarios os  valores maxi
mos e minimos para  o GRAFICO DE
 BARRAS"'': INPUT z$
2040 IF NOT z$="s" THEN GO TO 72
0
2060 PRINT INK 1;"QUER O GRAFICO
 PELOS VALORES     DE x OU y ? (x
/y)                 ""ENTER"" regr
essar": INPUT z$: CLS
2070 IF z$="x" THEN GO TO 2095
2080 IF z$="y" THEN GO TO 2500
2085 IF z$="" THEN GO TO 720
2090 GO TO 2060
2095 PRINT FLASH 1;"Espere um mo
mento."'': FLASH 0;"Pode demorar
um pouco!"
2096 REM VALORES DE X
2100 LET minx=b(a): LET maxx=b(1
): DIM r(15): DIM l(16): LET div
x=(maxx-minx)/15
2110 FOR b=1 TO 15: FOR n=1 TO a
2120 LET p9=0: LET r9=minx+divx
2125 IF b=15 THEN LET r9=r9+.000
001
2130 IF x(n)>=minx AND x(n)<r9 T
HEN LET p9=p9+1: LET r(b)=r(b)+p
9: LET l(b)=l(b)+p9
2140 NEXT n
2150 LET minx=minx+divx
2160 NEXT b
2165 PRINT "Aguenta..!"
```

```
2170 FOR j=1 TO 15: FOR k=1 TO 15
2180 LET l=0: IF l(k+1)>l(k) THEN LET l=l(k): LET l(k)=l(k+1): LET l(k+1)=l
2190 NEXT k: NEXT j
2200 LET highx=l(1): LET scalex=1
2210 IF highx>18 THEN LET scalex=scalex+1: LET highx=highx-18: GO TO 2210
2250 GO TO 3000
2500 PRINT FLASH 1;"Espere um momento..": FLASH 0;"Pode demorar um pouco!"
2505 REM VALORES DE y
2510 LET miny=c(a): LET maxy=c(1): DIM s(15): DIM m(16): LET divy=(maxy-miny)/15
2520 FOR b=1 TO 15: FOR n=1 TO a
2530 LET p9=0: LET r9=miny+divy
2535 IF b=15 THEN LET r9=r9+.000001
2540 IF y(n)>=miny AND y(n)<r9 THEN LET p9=p9+1: LET s(b)=s(b)+p9: LET m(b)=m(b)+p9
2550 NEXT n
2560 LET miny=miny+divy
2570 NEXT b
2575 PRINT : PRINT AT 10,10;"Aguenta..!"
2580 FOR j=1 TO 15: FOR k=1 TO 15
2590 LET l=0: IF m(k+1)>m(k) THEN LET l=m(k): LET m(k)=m(k+1): LET m(k+1)=l
2595 PRINT : PRINT AT 15,10;"Agora vai..!"
2600 NEXT k: NEXT j
2610 LET highy=m(1): LET scaley=1
2620 IF highy>18 THEN LET scaley=scaley+1: LET highy=highy-18: GO TO 2620
2630 GO TO 3500
3000 REM DRAW BAR CHART DE x
3010 IF scalex=1 THEN GO TO 3030
3020 FOR b=1 TO 15: LET r(b)=INT (r(b)/scalex+.5): NEXT b
3030 BEEP .25,20: CLS : DIM j$(18): PRINT "Nome do eixo horizontal de (x): ": INPUT i$
3040 PRINT "Nome do eixo vertical de (y): ": INPUT j$: CLS
3050 BEEP .8,30: PLOT 16,16: DRAW 0,143: PLOT 16,16: DRAW 239,0: PRINT AT 0,4;"GRAFICO DE BARRAS PARA x": PRINT AT 19,1;"0": PRINT AT 20,2;b(a): PRINT AT 20,28;b(1): PRINT AT 21,6;i$: FOR z=0 TO 17: PRINT AT z+4,0;j$(z+1): NEXT z: PRINT AT 2,0;18*scalex
3055 LET sp=2
3060 FOR b=1 TO 15
3065 IF r(b)=0 THEN LET sp=sp+2: GO TO 3110
3070 FOR j=1 TO r(b)
3080 PRINT INK 2;AT 20-j,sp;"█": PRINT INK 2;AT 20-j,sp+1;"█"
3090 NEXT j
3100 LET sp=sp+2
3110 NEXT b
3150 GO SUB 650: CLS : GO TO 2060
3500 REM GRAFICO DE BARRAS PARA y
3510 IF scaley=1 THEN GO TO 3530
3520 FOR b=1 TO 15: LET s(b)=INT (s(b)/scaley+.5): NEXT b
3530 BEEP .25,20: CLS : DIM k$(18): PRINT "Nome do eixo horizontal do (x): ": INPUT i$
3540 PRINT "Nome do eixo vertical do (y): ": INPUT k$: CLS
3550 BEEP .8,30: PLOT 16,16: DRAW 0,143: PLOT 16,16: DRAW 239,0: PRINT AT 0,6;"GRAFICO DE BARRAS PARA y": PRINT AT 19,1;"0": PRINT AT 20,2;c(a): PRINT AT 20,28;c(1): PRINT AT 21,6;i$: FOR z=0 TO 17: PRINT AT z+4,0;k$(z+1): NEXT z: PRINT AT 2,0;18*scaley
3560 LET sp=2
3570 FOR b=1 TO 15
3580 IF s(b)=0 THEN LET sp=sp+2: GO TO 3630
3590 FOR j=1 TO s(b)
3600 PRINT INK 2;AT 20-j,sp;"█": PRINT INK 2;AT 20-j,sp+1;"█"
3610 NEXT j
3620 LET sp=sp+2
3630 NEXT b
3650 GO SUB 650: CLS : GO TO 2060
5020 IF y$="n" OR y$="N" THEN GO TO 720
6000 REM SAIDA
6010 PRINT "Tem a certeza ? (s/n)": INPUT y$: CLS
6030 IF y$="s" OR y$="S" THEN GO TO 6050
6040 GO TO 6010
6050 PRINT "Quer regressar ao programa ?    (s/n)": INPUT y$: CLS
6060 IF y$="s" OR y$="S" THEN RESTORE 600: GO TO 70
6070 IF y$="n" OR y$="N" THEN BEEP .9,20: PRINT INK 2;"O.K. ADEUS!": STOP
6080 GO TO 6050
7000 REM SOMA DOS TOTAIS MEDIA, DP, CV, RPM, REG, COR, GL & t
7010 LET t1=0: LET t2=0: LET t3=0: LET t4=0: LET t5=0: LET s1=0: LET s2=0
7020 FOR h=1 TO a
7030 LET t1=t1+x(h): LET t2=t2+y(h): LET t3=t3+x(h)↑2: LET t4=t4+y(h)↑2: LET t5=t5+x(h)*y(h)
7040 NEXT h
7050 LET q=t2/a: LET q=INT (q*10↑3+.5)/(10↑3): REM MEDIA (y)
7060 LET p=t1/a: LET p=INT (p*10↑3+.5)/(10↑3): REM MEDIA (x)
7070 LET d=(t5-(t1*t2/(h-1)))/(t3-(t1↑2/(h-1))): REM REG(b)
7080 LET u=(t5-(t1*t2/(h-1)))/SQR ((t3-(t1↑2/(h-1)))*(t4-(t2↑2/(h-1)))): REM COR(r)
7090 LET e=q-d*p: REM REG(c)
7100 FOR i=1 TO a
7110 LET s1=s1+(x(i)-p)*(x(i)-p): LET s2=s2+(y(i)-q)*(y(i)-q)
7120 NEXT i
7130 LET s3=SQR (s1/(i-2)): LET s4=SQR (s2/(i-2)): REM DP(x&y)
7140 LET v1=s3/p*100: LET v2=s4/q*100: REM CV(x&y)
7150 LET s5=s3/SQR (i-1): LET s6=s4/SQR (i-1): REM EPM(x&y)
7160 LET u2=2*a-2: REM GL
7170 IF q>p THEN LET t=(q-p)/SQR ((2/(u2*a))*(t3+t4-(t1↑2/a)-(t2↑2/a))): GO TO 7190
7180 LET t=(p-q)/SQR ((2/(u2*a))*(t3+t4-(t1↑2/a)-(t2↑2/a))): REM t-TEST
7190 RETURN
```

DESCRICAO DO PROGRAMA

Linhas 25-190
Leitura dos Graficos, Entrada de dados (X,Y)

Linhas 200-595
Alterar entradas incorrectas, recomposicao dos dados

Linhas 600-630
"DATA" Graficos

Linhas 700-710
Calculo dos valores estatisticos

Linhas 720-840
Menu

Linhas 900-995
Estatistica descritiva (Media, Desvio Padrao, Coeficiente de variacao, Erro Padrao, f-Teste

Linhas 1000-1090
Impressao de dados

Linhas 1100-1195
Ordenacao descendente dos dados

Linhas 1200-1290
Medias Minimas e Maximas

Linhas 1300-1390
Analise Estatistica-Regressao

Linhas 1400-1470
Coeficiente de correlacao

Linhas 1500-1580
t-TESTE, teste de significancia

Linhas 1600-1800
Grafico Linear

Linhas 2000-3650
Grafico de barras

Linhas 6000-6080
Saida do programa

Linhas 7000-7200
Calculo dos valores estatisticos -MEDIA, C.V, D.P, E.P.M, F, t, G.L, r, b e c

O Z 80 — NOVAS INSTRUÇÕES

In "Micro-Systemes" — Jan./Fev. 82, pág. 80

As "novas" operações do Z 80 incidem essencialmente nos registos de índices IX e IY, executando registos sem quaisquer restrições, que completam harmoniosamente as instruções já existentes.

Segundo a sua categoria, agrupamos os diferentes códigos em várias tabelas.

O quadro 1 descreve as operações para carregar um registo de 8 bits. Adoptamos as seguintes convenções:

Bytes menos significativos:
IXL = Índice X "LOW"
IYL = Índice Y "LOW"

Bytes mais significativos:
IXH = Índice X "High"
IYH = Índice Y "High"

REGISTO "FONTE"												
	A	B	C	D	E	H	L	IXH	IXL	IYH	IYL	CARGA IMED.
A	7F	78	79	7A	7B	7C	7D	DD7C	DD7D	FD7C	FD7D	3E
B	47	40	41	42	43	44	45	DD44	DD45	FD44	FD45	06
C	4F	48	49	4A	4B	4C	4D	DD4C	DD4D	FD4C	FD4D	0E
D	57	50	51	52	53	54	55	DD54	DD55	FD54	FD55	16
E	5F	58	59	5A	5B	5C	5D	DD5C	DD5D	FD5C	FD5D	1E
H	67	60	61	62	63	64	65					26
L	6F	68	69	6A	6B	6C	6D					2E
IXH	DD67	DD60	DD61	DD62	DD63			DD64	DD65			DD26
IXL	DD6F	DD68	DD69	DD6A	DD6B			DD6C	DD6D			DD2E
IYH	FD67	FD60	FD61	FD62	FD63					FD64	FD65	FD26
IXL	FD6F	FD68	FD69	FD6A	FD6B					FD6C	FD6D	FD2E

QUADRO 1 — As diferentes instruções "suplementares" possibilitam a transferência do conteúdo de qualquer registo para um outro registo do Z 80. Notem-se também os códigos operatórios correspondentes à sua carga imeddiata

O quadro 2 contém as instruções de extensão das diversas operações com 8 bits.

A	B	C	D	E	H	L	IXH	IXL	IYH	IYL
ADD 87	80	81	82	83	84	85	DD84	DD85	FD84	FD85
ADC 8F	88	89	8A	8B	8C	8D	DD8C	DD8D	FD8C	FD8D
SUB 97	90	91	92	93	94	95	DD94	DD95	FD94	FD95
SBC 9F	98	99	9A	9B	9C	9D	DD9C	DD9D	FD9C	FD9D
AND A7	A0	A1	A2	A3	A4	A5	DDA4	DDA5	FDA4	FDA5
XOR AF	A8	A9	AA	AB	AC	AD	DDAC	DDAD	FDAC	FDAD
OR B7	B0	B1	B2	B3	B4	B5	DDB4	DDB5	FDB4	FDB5
CP BF	B8	B9	BA	BB	BC	BD	DDBC	DDBD	FDBC	FDBD
INC 3C	04	0C	14	1C	24	2C	DD24	DD2C	FD24	FD2C
DEC 3D	05	0D	15	1D	25	2D	DD25	DD2D	FD25	FD2D

QUADRO 2 — As extensões permitem certas operações em 8 bits com um Z 80

Por analogia com a instrução SLA, que permite a deslocação de um "0" em qualquer registo, a instrução SLI pode ser definida — possibilita a deslocação de um "1" e afecta o registo do mesmo modo que SLA. Os códigos operatórios respectivos estão no quadro 3

	A	B	C	D	E	H	L	<HL>	<IX+d>	<IY+d>
SLA	CB 27	CB 20	CB 21	CB 22	CB 23	CB 24	CB 25	CB 26	DDCB d 26	FDCB d 26
SLI	CB 37	CB 30	CB 31	CB 32	CB 33	CB 34	CB 35	CB 36	DDCB d 36	FDCB d 36

QUADRO 3 — Os diferentes códigos operatórios das instruções SLA e SLI

A extensão e o tempo de execução das insruções "não oficiais" do Z 80 apresentam-se no quadro 4

INSTRUÇÕES	NÚMERO DE BYTES	NUMERO DE CICLOS	NÚMERO MICROCICLOS
SLI	IDENTICO A SLA		

LD r_1, r
LD r, r_1
LD r_1, r_1
	2	2	8

ADD A, r_1 CP A, r_1 INC r_1 DEC r_1	2	2	8

QUADRO 4

Observe-se que **r** pode ser um dos registos A, B, C, D ou E e que **r1** pode relativamente a ele representar IXH, IXL, IYH ou IYL.

STARS WAR

RESOLUÇÃO DOS GRÁFICOS

(V. n.º 15/Dezembro, pág. 10)

Vários leitores nos interrogaram como definir os gráficos do programa STARS WAR.
Só estarão definidos depois de ter introduzido as linhas 9 a 170.
Deve então fazer "RUN" e seguir o esquema abaixo indicado:
GRAPHICS — CAPS SHIFT + TECLA 9 em simultâneo que fará aparecer o cursor [G]
Agora utilize as teclas indicadas para cada gráfico:
— Linhas 270,500 e 540 — GRAPHICS + teclas E e F.
— Linha 280 e 520 — GRAPHICS + teclas A,B e C,D.
— Linha 9010 — GRAPHICS + teclas J, K.
— Linha 9700 — GRAPHICS + teclas I,I,H,H e GG.

NOVOS PROGRAMAS

SPECTRUM

— JOGOS E UTILITÁRIOS

- VALHALLA — 6 aventuras c/ dezenas de personagens nas quais se procuram objectos escondidos — **600$00**
- HUNTER THE KILLER — Aventura num submarino — **600$00**
- LUNAR JET MAN — Aventura espacial — **400$00**
- C. FLAG — Corrida de automóveis com opção de carro e pista — **400$00**
- CYRUS CHESS — Jogo de xadrez com grande capacidade de resposta (tempo) — **400$00**
- GAMÃO — Prática deste jogo — **400$00**
- DOMINÓ — Prática deste jogo — **400$00**
- MOLAR MAUL — Escovar os dentes de modo a não serem atacados pela cárie — **400$00**
- XADOM — Aventura espacial — **400$00**
- KEY FILE — Ficheiro com opções de formatação da ficha, procurar, alterar, limpar, gravar, carregar e imprimir (adaptado para Microdrive) — **800$00**
- MASTER FILE — Ficheiro adaptado para Microdrive c/ 32 K bytes disponíveis — **1 000$00**

— COMPILADORES DE LINGUAGEM FORTH

- ABERSOFT 48 K SPECTRUM FORTH — Compilador de linguagem FORTH que também possui editor. Todas as palavras Standard da linguagem FORTH estão incluídas e existem também operadores de dupla precisão.
  As extensões especificas para o SPECTRUM incluem instruções do tipo (gráfico) DRAW e ATTR e são sinónimas das instruções que existem em BASIC. Os gráficos definidos pelo utilizador são facilmente construídos.
  Existem duas formas de definir o SAVE, mas ambas diferentes do habitual: a primeira é uma rotina para gravar o núcleo das definições FORTH. O segundo método envolve 10 ecrans (screens) FORTH. Cada um pode ser compilado individualmente mas têm de ser gravados os 10 em conjunto.
  O editor de ecran apresenta 16 linhas de 64 caracteres.
  (Preço ainda não previsto)
- FLOATING POINT FORTH (48 K) — Possui 79 estruturas FORTH, funções trigonométricas e usa a impressora TIMEX ou SINCLAIR. Tem todos os gráficos de alta resolução e cor do Spectrum: PLOT, DRAW e BEEPER. Permite entrada directa de código máquina. — **600$00**

— ENGENHARIA CIVIL (48 K)

- CÁLCULO DE ESFORÇOS E DESLOCAMENTOS EM ESTRUTURAS CONTÍNUAS PLANAS
  Quem pretender examinar as folhas descritivas para ajuizar do grau de interesse deste programa, deverá remeter-nos 50$00 em selos do correio. — **4 000$00**

DESCONTO DE 20% PARA SÓCIOS DO CLUBE

LIVROS SPECTRUM *

Recebemos da LANDRY uma remessa de livros (10 de cada título) que estão à vossa disposição: aquisição directa na sede do CLUBE ou pedidos à cobrança.

TÍTULOS	PREÇO
— EXPLORING SPECTRUM BASIC	495$00
— ZX SPECTRUM HARDWARE	427$50
— MASTER YOUR ZX MICRODRIVE	360$00
— THE ZX SPECTRUM AND HOW TO GET THE MOST FROM IT	360$00
— 20 BEST PROGRAMS FOR THE ZX SPECTRUM	234$00
— 60 GAMES AND APPLICATIONS FOR THE ZX SPECTRUM	180$00
— SPECTRUM MICRODRIVE BOOK	315$00
— 40 BEST MACHINE CODE ROUTINES FOR THE ZX SPECTRUM	360$00
— THE SPECTRUM POCKET BOOK } NOVOS	405$00
— SPECTRUM SPECTACULAR } NOVOS	405$00

ESTES PREÇOS SÃO OS DA LANDRY, PELO QUE NÃO PODEMOS FAZER O HABITUAL DESCONTO DE 10%.

NOVO LIVRO (em fotocópia):

— LE GRAND LIVRE DU ZX SPECTRUM (Versão francesa do "THE ZX SPECTRUM EXPLORED"), HARTNELL Tim, Eyrolles, Paris 1983.

Preço (Fotocópias): 550$00

* O preço das fotocópias do livro LA CONDUITE DU ZX SPECTRUM é 560$00 e não 864$00 como referido no n.º anterior.

SINCLAIR QL: A ÚLTIMA NOVIDADE DA SINCLAIR

128 K RAM (possível de expandir até 640 K) • Processador de 32 bits (Motorola 68008) • Fornecido completo com 4 programas: processamento de texto, grelha de cálculo (tipo VISICALC), base de dados e gráficos comerciais • 2 Microdrives incorporados • Capacidade de utilização de rede de comunicações • Teclado profissional QWERTY • Sistema operativo QDOS com possibilidade de multiprocessamento • Monitor a cores ou TV • Interface RS232 incorporado e aceitação de Joysticks • Linguagem incorporada super BASIC.

PREÇO PREVISTO: 80.000$00

CLUBE Z80

INSCRIÇÃO COMO ASSOCIADO

O **CLUBE Z80** está aberto a todos os utilizadores de microcomputadores.

A intenção de associar os entusiastas das micro-máquinas, é exclusivamente a de permitir:

1 — PUBLICAÇÃO DE UM JORNAL MENSAL, onde sejam publicados programas de uso geral ou específico como no caso da educação.

2 — PROMOVER TROCAS DE PROGRAMAS, e trocas de experiências; tanto no caso do Software (programação), como no caso do Hardware (electrónica).

3 — PROMOVER DESCONTOS NA AQUISIÇÃO DE PROGRAMAS.

4 — LANÇAR CURSOS DE PROGRAMAÇÃO EM BASIC — PASCAL OU OUTRAS LINGUAGENS E DIVULGAR O USO DE LINGUAGEM MÁQUINA.

NOME ....

IDADE .... COMPUTADOR TIPO ....

PROFISSÃO ....

ENDEREÇO ....

.... TELEF. ....

ASSINATURA ANUAL — Esc. 1 500$00 ☐

ASSINATURA SEMESTRAL — Esc. 750$00 ☐

CHEQUE OU VALE DO CORREIO

N.º ....

BANCO ....

DATA ..../......./.......

JÁ SÓCIO ☐

NOVO SÓCIO ☐ → A partir do mês de .... (inclusive)

AS RAZÕES DO ATRASO NA SAÍDA DESTE NÚMERO

CAROS AMIGOS

O CLUBE Z8Ø lamenta profundamente que o nº de Janeiro só seja recebido por vós em Março. E nosso dever assumir também este facto apesar das razões nos serem completamente alheias:
Devido a sobrecarga de trabalho, a Tipografia que executa a n/ obra gráfica apenas nos conseguiu entregar este nº em Março.
Como se isso não bastasse, todos os exemplares saíram com péssima impressão das listagens, tal como demontramos ao lado.
Assim, fomos forçados a pedir uma reimpressão que originou nova demora.

Aqui ficam as n/ desculpas e um apelo à v/ compreensão.

Os Coordenadores

Alexandre Sousa
J. Magalhães
Maria Irene

SLOW

RUN...